

## **EXHIBIT H**

## 5.0 Reference

This section describes the actual OLE Interfaces exposed, the definitions of the data structures used when passing data around, and the definitions of each class used internally by the XMC Driver Administrator Component.

### 5.1 Structures and Defines

This section defines all structures, enumerations, and defines used by the driver.

#### 5.1.1 XMC\_DRIVER\_INFO Structure

The following structure is used when setting up and querying the state of the driver.

```
struct XMC_DRIVER_INFO
{
    XMC_HDRIVER           {in,out}    m_hDriver;
    XMC_DRIVER_MODULETYPE {out}        m_mt;
    BOOL                  {out}        m_bStreamMgmtLocked;
    LPTSTR                 {out}        m_pszHardwareVendor;
    DWORD                  {in}         m_cbMaxHardwareVendor;
    LPTSTR                 {out}        m_pszHardwareModel;
    DWORD                  {in}         m_cbMaxHardwareModel;
    LPTSTR                 {out}        m_pszDriverVendor;
    DWORD                  {in}         m_cbMaxDriverVendor;
};
```

#### 5.1.2 XMC\_STREAM\_INFO Structure

This structure is used to pass all stream specific data used to both setup the stream, and query the stream for its current settings.

```
struct XMC_STREAM_INFO
{
    XMC_HSTREAM           m_hStream;
    XMC_STREAM_MODULETYPE m_mt;

    union {
        struct PCBus
        {
            DWORD dwPort;
            DWORD dwIRQ;
        },

        struct Serial
        {
            DWORD dwPort;
            DWORD dwBPS;
        },

        struct TextFile
        {
            LPCTSTR pszFileName;
        },

        struct Custom
        {
            DWORD dwParam1;
            DWORD dwParam2;
        },
    };
};
```

### 5.1.3 XMC\_DRIVERINTERFACESUPPORT\_INFO

This structure defines all interface support information used to determine the level of XMC SPI supported by the driver..

```
struct XMC_DRIVERINTERFACESUPPORT_INFO
{
    TCHAR                m_szDriverName[ XMC_DRIVERNAME_MAX+1 ];
    DWORD                m_dwInterfaceCount;
    XMC_INTERFACESUPPORT_INFO m_rgIFSI[];
};
```

### 5.1.4 XMC\_INTERFACESUPPORT\_INFO

This structure defines the support information for one interface exposed by an XMC Driver.

```
struct XMC_INTERFACESUPPORT_INFO
{
    TCHAR                m_szInterfaceName[ XMC_INTERFACENAME_MAX+1 ];
    DWORD                m_dwMethodCount;
    XMC_METHODSUPPORT_INFO m_rgMSI[];
};
```

### 5.1.5 XMC\_METHODSUPPORT\_INFO

This structure defines the support information for one method in an interface exposed by an XMC Driver.

```
struct XMC_METHODSUPPORT_INFO
{
    TCHAR                m_szMethodName[ XMC_METHODNAMEMAX+1 ];
    XMC_SUPPORTTYPE      m_st;
};
```

### 5.1.6 XMC\_SUPPORTTYPE

This enumeration defines the different types of support provided for a certain method.

```
enum XMC_SUPPORTTYPE
{
    XMC_ST_DRIVER,
    XMC_ST_STUB
};
```

## 5.4 Classes

This section contains the definition of all classes used by the XMC Driver Administrator Control Panel Applet in its implementation.

### 5.4.1 CDriverAdmin Class

The CDriverAdmin class is used to directly communicate with the driver administrator component. All OLE related details are encapsulated within this class. The following is the definition of the CDriverAdmin class.

```
class CDriverAdmin
{
public:
    //---- Constructors & Destructors ----

    CDriverAdmin( void );
    ~CDriverAdmin( void );

    //---- Initialization ----
```

```

        DWORD Initialize( void );

        //----- Actions -----

        DWORD EnumDriver( LPENUMDRIVER *ppEnumDriver );

        DWORD RegisterDriver( LPCTSTR pszFileName,
                               BOOL bEnabled,
                               LPXMC_HDRIVER phDrv );
        DWORD RegisterStream( LPCTSTR pszFileName,
                               BOOL bEnabled,
                               XMC_HDRIVER hDrv,
                               LPXMC_HSTREAM phStrm );
        DWORD UnRegister( XMC_HDRIVER hDrv, XMC_HSTREAM hStrm );

        DWORD GetSupportInfo( LPXMC_DRIVERINTERFACESUPPORTINFO rgDIFS );

        //----- Debugging -----

        DWORD EnableDiagnostics( BOOL bEnable );

    private:
        //----- Private Data -----

        LPXMC_DRIVERADMIN      m_lpDA;
        LPXMC_DRIVERADMINDEBUG m_lpDADbg;
};

```

### 5.4.2 CDriverInfoMap Class

The CDriverInfoMap class describes the current state of the driver administrator component regarding installed drivers and streams. The driver administrator CPL queries the driver administrator component for this block of information, that it then uses to update the user-interface describing the drivers and streams installed.

```

class CDriverInfoMap
{
    public:
        //----- Constructors & Destructors -----

        CDriverInfoMap( void );
        ~CDriverInfoMap( void );

        //----- Actions -----

        DWORD Load( LPUNKNOWN *rgpDrvUnk, DWORD dwDrvCount,
                    LPUNKNOWN *rgpStrmUnk, DWORD dwStrmCount );

        //----- Public Data -----

        LPXMC_DRIVERINFO      m_rgDriverInfo{};
        DWORD                 m_cbDriverInfo;
};

```